基于 ROS 的 ICP-SLAM 在嵌入式移动机器人上的实现与优化 *

郑富瑜，何　苗†，张　辉，赵韦人，熊德平，梁安辉，王成民，王　润，周海亮

(广东工业大学 物理与光电工程学院，广州 510006)

摘　要： 基于激光雷达的 ICP-SLAM 在机器人操作系统中无法直接运行，且通过 PC 实现算法时，没有高效运载计算机资源，产生资源浪费，同时降低了移动机器人系统的灵活性。通过在嵌入式系统中搭建机器人操作系统，在系统中移植 MRPT 函数库中的 ICP-SLAM 算法，对算法进行优化，调整算法运行的 CPU 占用率，使用基于激光雷达的移动机器人进行建图。结果表明机器人操作系统生成的栅格地图可以满足机器人自主导航的需求，SLAM 效果更为直观，系统更加灵活，成本和硬件配置要求大大降低。证明优化后的算法更贴近实际的使用需求。

关键词： ICP-SLAM；ROS；移动机器人；嵌入式系统

中图分类号： TP242　　doi: 10.3969/j.issn.1001-3695.2017.12.0820

Implementation and optimization of ROS based ICP-SLAM embedded robot

Zheng Fuyu, He Miao†, Zhang Hui, Zhao Weiren, Xiong Deping, Liang Anhui,
Wang Chengmin, Wang Run, Zhou Hailiang

(*School of Physics & Optoelectronic Engineering Guangdong University of Technology*, *Guangzhou* 510006, *China*)

Abstract: Laser-based ICP-SLAM cannot run directly in the robot operating system, and there is no efficient carrying of computer resources when implementing the algorithm on PC, resulting in waste of resources, while reducing the flexibility of mobile robot system. Build robot operating system on the embedded system, transplant the ICP-SLAM algorithm of the MRPT library into the system, optimize and fixthe CPU occupancy rate of the algorithm, test the function of SLAM by the mobile robot based on the LIDAR. The results show that the grid map thatrobot operating system generated have met the requirements of robot autonomous navigation, and theeffect of SLAM is more intuitive, besides, the system is more flexible, greatly reducecost and hardware configuration requirements. Proved that the optimized algorithm is closer to the practical requirements.

Key words: ICP-SLAM; ROS; mobile robot; embedded system

0　引言

即时定位与构建地图（SLAM）技术是移动机器人实现自主导航的关键，在未知环境中，通过传感器对周围进行探测，构建出环境地图，对机器人进行定位，从而结合传感器生成的实时地图进行导航。在 SLAM 的研究过程中，使用视觉传感器的方式容易受到环境光的干扰，生成的数据量巨大，限制了传感器的应用环境，同时提高了数据处理的成本[1]。移动机器人进行 SLAM 时，通过传感器收集到的数据还原环境分布，使用算法求解机器人的位姿[2]。ICP 算法由 Besl 和 McKay 于 1992 年提出，是一个快速且精确的算法，在图像和点集的配准中得到广泛的应用[3]。由于初值不佳或者在扫描匹配过程中出现误差，ICP 算法在可能会收敛到局部最优点上，提高了算法对于传感器的要求[4]。ICP-SLAM 算法将新的测距扫描结果与上一次结果生成的点云图进行匹配，从而增量地构建地图[5]。点云图随着时间的推移计算量持续增大，增加了系统的负担[6]。ICP-SLAM 需要比特征提取更多的计算量，在 PC Linux 系统中使用 ICP-SLAM，结合高分辨率的激光雷达可以较好的实现高精度的 SLAM 效果，但使用 PC 相当于提高硬件配置，且只占用 PC 很小一部分的功能，造成功能浪费和成本增加[7]。但移动机器人的普通嵌入式开发板配置不够，无法实现高精度的地图构建和机器人实时定位。MRPT 库中的 ICP-SLAM 没有在 ROS 中的节点，无法使用 ROS 评估工具进行仿真平台的验证。

针对以上 ICP-SLAM 无法在 ROS 中运行，硬件配置要求

收稿日期： 2017-12-04；**修回日期：** 2018-01-30　**基金项目：** 广东省科技计划资助项目（2014B050505020，2014B090904045，2015B010114007，2016B010126005）；广州市科技计划资助项目（2016201604030027）

作者简介： 郑富瑜（1994-），男，广东陆丰人，硕士研究生，主要研究方向为移动机器人、激光 SLAM 算法；何苗，男（通信作者），博导，主要研究方向为 LED 封装（herofate@126.com）；张辉，男，博士（后），主要研究方向为微纳光学；赵韦人，男，浙江人，教授，主要研究方向为固态照明技术；熊德平，男，江西人，副教授，主要研究方向为 LED 封装及应用；梁安辉，男，教授，主要研究方向为高速光通信系统、模块及器件；王成民，男，山西人，硕士研究生，主要研究方向为 LED 封装；王润，男，广东梅州人，硕士研究生，主要研究方向为嵌入式应用；周海亮，湖北武人，硕士研究生，主要研究方向为嵌入式应用.

对于移动机器人普通嵌入式不适用的问题，本文将 MRPT C++库中的 ICP-SLAM 移植在移动机器人嵌入式开发板上，保留算法的部分源码并作出修改，优化地图构建中的耗时与计算资源的部分，增加基于栅格地图实时轨迹显示部分等。移植后的 ICP-SLAM 大大降低了对硬件平台的要求，提高了系统的灵活性，显示效果更为直观，还添加了实时轨迹的显示，提升了 SLAM 的性能，精确反映了周围环境的布局。

1　算法优化与改进

1.1　总体流程设计

本文将 PC 端的 MRPT C++库中的 ICP-SLAM 进行移动机器人嵌入式 ROS 平台中的移植和优化，最终在显示终端显示基于 ROS 平台的包含运动轨迹的栅格地图，即地图的实时构建。算法可以分成以下几个步骤：完成选择控制点，扫描匹配，通过最小化匹配点误差方程求出旋转矩阵与平移矩阵，即时定位和地图构建。总体的系统设计流程如图 1 所示。选择控制点在激光雷达与里程计部分完成，扫描匹配之后，进行迭代求出旋转、平移矩阵，不断迭代，求解机器人的运动，完成即时定位和地图构建。

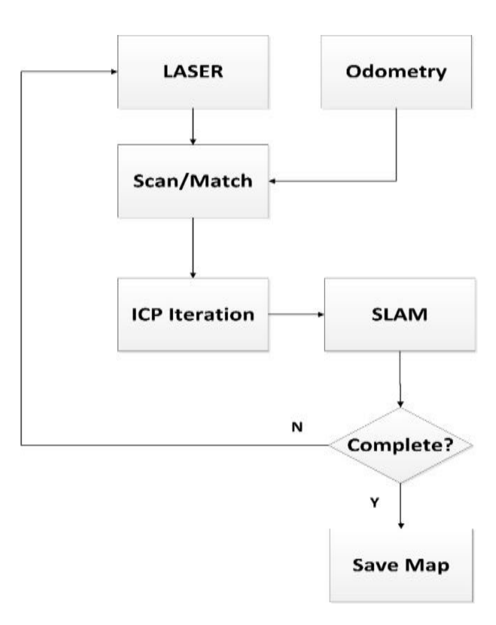


图 1　总体设计流程

在上述系统设计流程中，本文通过修改 ICP-SLAM 算法源代码部分的 C++代码，同时删去了与本文 ICP-SLAM 方法不同的源代码，包括 MRPT C++库中的 EKF、RBPF 和 Graph-slam 方法；在地图的匹配方式与方法上进行改进，优化算法对于 CPU 需求的计算量；在 ROS 的图形显示工具中进行 SLAM，并添加路径的显示。

1.2　ICP-SLAM 源代码

ICP-SLAM 算法的主要在 src 和 include 文件夹下实现，在 mrpt_icp_slam_2d_wrapper.cpp 中使用栅格地图数据 occu_msg 替换点云地图数据；对匹配点搜索使用的方法进行修改，将 icp-slam-live.ini 文件中的匹配方法 ICP_algorithm 改为 grid map，使用栅格图进行扫描匹配激光雷达得到的环境地图；在 mrpt_icp_slam_2d_wrapper.cpp 加入读取里程计的读数 odom 的函数，再对里程计的读数进行四元数的转换，为扫描匹配初期 ICP 算法提供初值姿态矩阵，ICP 算法加入机器人里程计提供初始姿态矩阵，大大减少初始时刻全局搜索带来的时间消耗。在 mrpt_icp_slam_2d_wrapper.cpp 中删去 3D 显示窗口 3DWindows，关闭同时定位与地图构建的实时显示，同时添加轨迹显示 updatetrajectory，采用基于 ROS 的图形工具 RVIZ 进行界面和轨迹的显示。

1.3　扫描匹配

进行扫描匹配时，最邻近点迭代（iterative closest point，ICP）算法通过最小化匹配误差求解旋转矩阵与平移矩阵，在 MRPT C++库中有多种不同的方法。算法对扫描得到的数据帧进行匹配、更新局部区域地图、调整局部地图偏差，由于缺少初始姿态矩阵，达到收敛条件需要的迭代次数大量增加。考虑到耗时且使用里程计提供初始姿态矩阵对于扫描匹配的精度影响较小，同时迭代次数减少降低了对嵌入式开发板 CPU 计算量的载荷，为嵌入式的引入提供契机。

ICP 算法在 MRPT C++库中有多种方法，主要包括经典点云匹配方法，经典 ICP 方法（Classic ICP）[8]，列文伯格-马夸尔特方法（Levenberg-Marquardt，LM）[9]、八叉树图方法（octomap）[10]和栅格地图方法（gridmatch）等[11]。本文将扫描匹配算法中点云匹配的方式调整为栅格地图匹配，避免了后期点云匹配长时间匹配产生的大量迭代计算，确保精确率的基础上降低对硬件的要求，更加适用于嵌入式开发板的配置。优化后的扫描匹配流程如图 2 所示。激光雷达采集到数据之后，通过对编码器为主要结构的里程计读数进行检测，机器人位移小于一定阈值时，不进行旋转平移匹配，沿用上一帧栅格图。

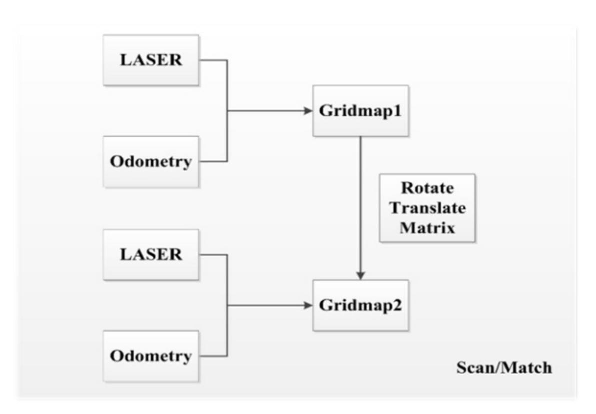


图 2　扫描匹配流程

1.4　ROS 显示与观测

机器人操作系统(ROS)是一个机器人软件操作平台，前身由斯坦福大学人工智能研究所(Stanford Artificial Intelligence Laboratory)所研发。ROS 可以建立节点管理器，从命令行客户端查看特定主题不断发布的消息数据[12]。

为了能够借助 ROS 的评估工具和功能，在嵌入式开发板上安装 Ubuntu 并搭载 ROS 平台，订阅激光雷达采集数据后发布

的 Laserscan 主题。在 ROS 中创建节点，移植 ICP-SLAM 算法，将源代码放置于 src 文件夹下，将 ini 可执行文件放置于 tutorial 文件夹中，创建 launch 文件同时驱动多个节点，包括订阅激光雷达发布的主题的节点、使用 ICP-SLAM 可执行文件进行建图的节点和使用 ROS 图像显示界面工具 RVIZ 进行实时显示建图过程。同时，在 RVIZ 的实时显示中添加机器人的运动轨迹。ROS 中主题的发布与订阅如图 3 所示。通过订阅移动机器人里程计发布的主题 odom 和激光雷达发布的主题 scan，ICP-SLAM 算法会对采集到的数据进行处理。

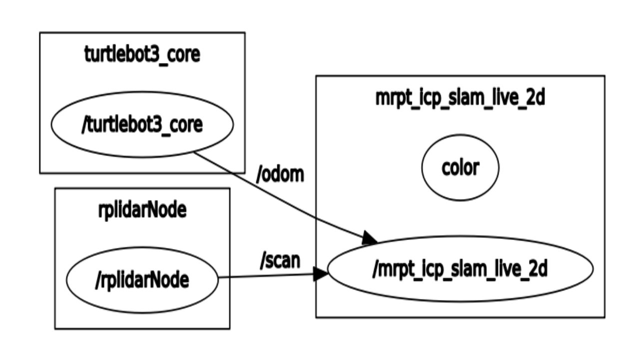


图 3　ROS 主题发布与订阅

2　算法的测试与结果解析

2.1　测试硬件配置

测试的硬件配置为台式计算机、激光雷达和自制的移动机器人底盘，将使用台式计算机运行算法与机器人采用嵌入式开发板独立运行算法的对比。机器人采用的激光雷达采用二维激光雷达，移动机器人底盘模块主要由上位机和下位机组成，上位机主要由移动机器人普通嵌入式开发板树莓派三代和 PC 组成，树莓派负责激光雷达获取数据的处理、ROS 软件上 ICP-SLAM 算法的运行和机器人移动命令的发布，PC 用于显示开发板处理之后的数据和图形界面，并未参与到算法的计算部分，仅发挥观测的作用。下位机主要由基于 Arduino Uno 的开发板、电机驱动板、集成磁电编码器的直流电机和电源组成，基于 Arduino Uno 的开发板主要负责机器人的移动功能实现，集成磁电编码器的直流电机负责产生里程计数据,机器人的上、下位机供电统一由电源提供。测试硬件配置实物图如图 4 所示，分别是机器人整体图、树莓派开发板与激光雷达。



图 4　硬件配置实物图

进行算法的测试时，个人电脑计算能力强，存储空间大，但运行 ICP-SLAM 时只用到小部分运算能力，对于移动机器人开发而言，个人电脑与机器人的捆绑开发不仅导致未使用的计算资源严重浪费，且提升了硬件配置成本和功耗，降低了系统的灵活性与集成性。嵌入式开发板以其低成本、低功耗、体积小和适当的 CPU 运算效率等优点，成为 SLAM 算法运行时个人电脑的替代品。表 1 所示为个人电脑与嵌入式开发板树莓派三代的硬件参数对比。除此之外，硬件的价格上，树莓派只需要 40 美元左右，价格约为 PC 的 10%。

表 1　硬件参数对比

	PC	Raspberry Pi
CPU	Intel(R) core(TM) i5-4660 CPU@3.20GHz	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
Storage	777GB	Micro SD card 32GB
Memory	4G DDR3 1600MHz	1GB

2.2　不同匹配算法仿真对比

使用不同的 ICP 匹配算法进行扫描匹配时，精确程度不尽相同，除了实时建图质量可以反映精度之外，还可以通过 ICP-SLAM 定位功能生成的轨迹来判断。不同算法产生的轨迹与真实值之间的误差不同，为了得到轨迹的真实值，使用 MRPT C++ 库 apps 文件夹下的模拟器，通过运行模拟器中的模型机器人，在模拟的环境中进行运动，得到运行轨迹的真实值文件和数据包，通过选择不同 ICP 匹配算法运行同一数据包，得到对应的轨迹文件，使用 MATLAB 仿真对比轨迹真实值与 MRPT C++ 库中几种不同的匹配算法产生的轨迹，包括气体映射方法（gas_mapping）、列文伯格-马夸尔特方法（Levenberg-Marquardt，LM）、八叉树图方法（octomap）和栅格地图方法（gridmatch）与真实值（Ground Truth）的轨迹，得到的轨迹图如图 5 所示。不同匹配方法与真实值的方差如表 2 所示。

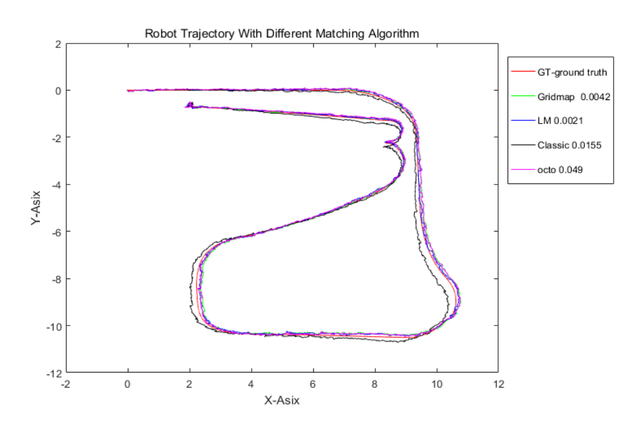


图 5　不同匹配方法轨迹图

表 2　不同匹配方法与真实值的方差

Matching Algorithm	Gridmap	LM	Classic	Octomap
Variance	0.0042	0.0021	0.0155	0.049

根据表 2，使用列文伯格-马夸尔特方法（LM）与真实值的方差最小，为 0.002 1，小于使用栅格地图方法的方差 0.004 2，但在嵌入式开发板运行过程中，使用 LM 方法的 CPU 的占用率远大于使用栅格地图，当 CPU 占用率过高时，算法的运行速度和数据处理速度会受到限制，即时建图容易产生漂移，导致地图的精度下降甚至无法建立完整的地图。表 3 所示为算法运行时嵌入式开发板运行的 CPU 占用率。由于嵌入式开发板的 CPU 计算资源十分宝贵，列文伯格-马夸尔特方法明显对 CPU 的消耗更大，当嵌入式开发板在移动机器人上使用时，复杂的环境会导致 CPU 的浮动更加明显，更低的 CPU 消耗让开发板的容错率更高，鲁棒性更好。

表 3 不同匹配方法嵌入式开发板的 CPU 占用率

Algorithm	Max	Average	Min
Gridmap	40.35%	38.76%	36.84%
LM	56.82%	49.84%	48.34%
Classic	50.77%	47.68%	45.31%
Octomap	66.58%	63.49%	62.44%

2.3 优化后的算法测试

测试场地为 7.8m x 15m 大小的办公区域，使用基于二维激光雷达的移动机器人底盘进行即时定位和建图。图 6 所示为机器人底盘 SLAM 生成的地图，黑色部分是障碍物，白色为可通行区域，绿色实线为机器人底盘的运动轨迹。对比图 7 使用嵌入式开发板运行 MRPT C++库中的 ICP-SLAM 生成的地图与图 8 使用 PC 作为运算单元运行 MRPT C++库 ICP-SLAM 生成的地图，图 6 使用嵌入式开发板运行改进后的 ICP-SLAM 算法的方式生成的地图可以明确辨认出障碍物的位置、环境的布局、机器人运动的轨迹，验证了使用嵌入式开发板的可行性与实用性。图 7 由于嵌入式开发板在运行时内存耗尽，容易产生地图漂移，导致即时建图的准确率降低，而图 8 由于 PC 硬件配置较高，建图时清晰度与对比度较高，并在运行中添加机器人模型，但该图生成时，算法选择点云与栅格图同时生成，让边缘部分厚度增加，且没有路径显示，不能够直观反映机器人的实时定位。

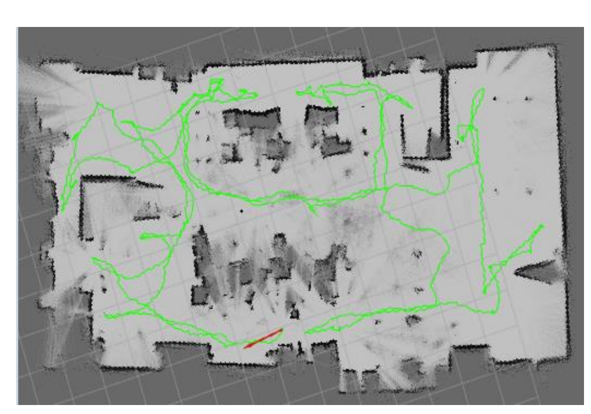


图 6 ICP-SLAM 构建地图

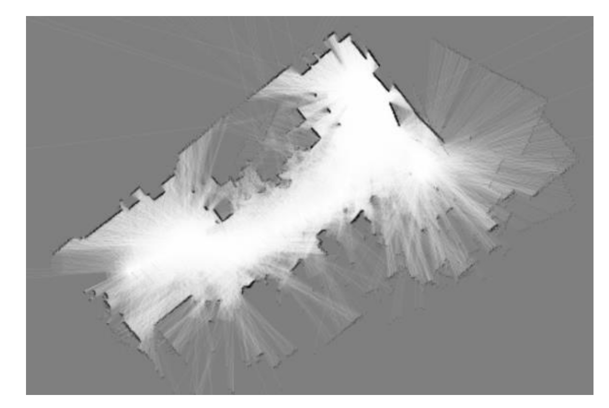


图 7 MRPT C++库 ICP-SLAM

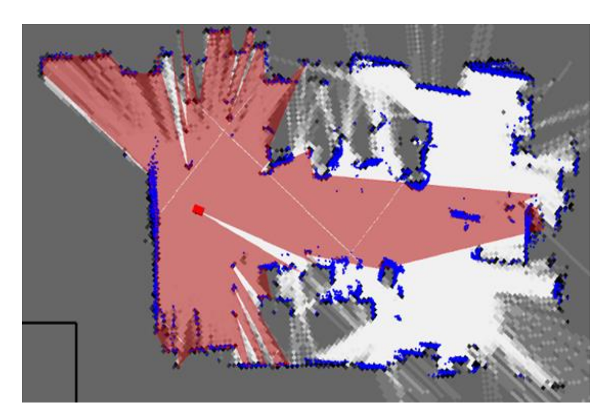


图 8 PC 运行 ICP-SLAM

2.4 总体改进效果

将移植后的 ICP-SLAM 在嵌入式开发板上运行与在系统中运行 MRPT C++库中的 ICP-SLAM 进行对比，同一个数据包，在嵌入式开发板系统中使用 MRPT C++库中原始的 ICP-SLAM 运行时，迭代的总次数为 27 349 次，运行时间为 15 min 37 s，使用优化后 ICP-SLAM 算法迭代次数为 2 954 次，运行时间为 5 min 29 s，CPU 占用率上，使用 MRPTC++库中原始的 ICP-SLAM 平均值达到 136%，优化后降低到了 38.76%。移植后的算法提高了运算速率，同时降低了 CPU 占用率，确保了运行时算法的测量准确率不会因 CPU 资源耗尽而产生下滑。

将生成的地图通过 ROS 节点 map_server 保存下来，移动机器人使用自适应蒙特卡洛定位（adaptive Monte Carlo localization，AMCL）算法，结合激光雷达进行室内的导航，导航效果如图 9 所示。

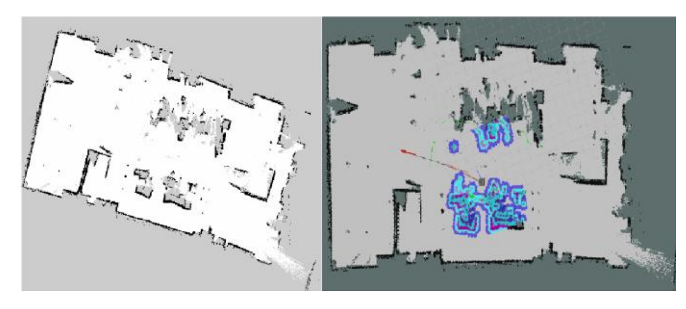


图 9 保存地图进行导航

3 结束语

本文提出并实现了将 ICP-SLAM 进行改进和优化，移植到移动机器人嵌入式开发板树莓派三代中，最终实现了 SLAM；同时算法的运行速度和 CPU 占用率两方面得到提升，大大降低了移动机器人实现 SLAM 硬件配置要求和设备的成本。针对嵌入式开发板运算速率有限导致 MRPT C++库中 ICP-SLAM 不适用的问题，在保证精确率的基础上，通过修改原点云图匹配方式为栅格图匹配，去除 ICP-SLAM 实时显示界面，降低了算法对 CPU 计算量的需求，缩短了算法运行耗时。关于构建地图过程中显示不够直观的问题，采用在 ROS 平台上使用 RVIZ 图像显示界面，结合自带评估工具在 PC 终端中进行显示；同时添加了路径显示功能，既符合现在的机器人研究趋势，又更加直观地反映了实际环境的布局。通过对算法的改进，从而提高了移动机器人的集成度和灵活度，降低了移动机器人的硬件成本。随着嵌入式技术的不断发展，嵌入式机器人的功能将不断延伸，普及生活中的更多领域。

参考文献：

[1] Kim J, Jeong J, Shin Y S, *et al.* LiDAR configuration comparison for urban mapping system [C]// Proc of the 14th IEEE International Conference on Ubiquitous Robots and Ambient Intelligence. 2017: 854-857.

[2] Mendez O, Hadfield S, Pugeault N, *et al.* SeDAR-semantic detection and ranging: humans can localise without LiDAR, can robots? [J]. arXiv preprint arXiv: 1709. 01500, 2017.

[3] Besl P J, McKay N D. Method for registration of 3-D shapes [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1992, 1611 (2): 239-256.

[4] Tiar R, Ouadah N, Azouaoui O, *et al.* ICP-SLAM methods implementation on a bi-steerable mobile robot [C]// Proc of the 11th IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics. 2013: 1-6.

[5] Ghorpade V K, Borrmann D, Checchin P, *et al.* Time-of-flight depth datasets for indoor semantic SLAM [J/OL]. [2017-12-01]. https://www. researchgate. net/publication/319464322_Time-of-flight_depth_datasets_for_indoor_semantic_SLAM.

[6] Georgiou C, Anderson S, Dodd T. Constructing informative Bayesian map priors: a multi-objective optimisation approach applied to indoor occupancy grid mapping [J]. The International Journal of Robotics Research, 2017, 36 (3): 274-291.

[7] Mur-Artal R, Tardós J D. Probabilistic semi-dense mapping from highly accurate feature-based monocular SLAM [M]// Robotics: Science and Systems. 2015.

[8] Thorpe M. Gas mapping LiDAR for large-area leak detection and emissions monitoring applications [M]// CLEO: Applications and Technology. Optical Society of America. 2017.

[9] Li Haoran, Zhang Qichao, Zhao Dongbin. Comparison of methods to efficient graph slam under general optimization framework [C]// Proc of the 32nd Youth Academic Annual Conference of Chinese Association of Automation. 2017: 321-326.

[10] Vlaminck M, Luong H, Philips W. Multi-resolution ICP for the efficient registration of point clouds based on octrees [C]// Proc of the 15th IAPR International Conference on Machine Vision Applications. 2017: 334-337.

[11] Sasongko D F, Miura J. An integrated exploration and observation planning for an efficient indoor 3D mapping [C]// Proc of IEEE International Conference on Mechatronics and Automation. 2017: 1924-1929.

[12] Garage W. Robot operating system (ROS) [Z]. 2012.